# Signalogic VDI Installation Guide

**Includes Installation for**

- **Server Software**
- **Remote Device Software**

**Copyright © 2015 Signalogic, Inc.**

**Revision 1**

**Jun 2015**

**Signalogic, Inc.**
**9617 Wendell @ Skillman**
**Dallas, TX 75243**
**Tel: 214-349-5551**
**Fax: 214-343-0163**

**http://www.signalogic.com**
**tech_support@signalogic.com**

# Table of Contents


1 VDI INSTALLATION OVERVIEW 1

2 SERVER SOFTWARE INSTALLATIONS 2

2.1 INSTALLING HOST LINUX 2
2.1.1 INSTALLING LIGHTWEIGHT GUI FOR UBUNTU 14.04 2
2.1.2 INSTALLING CENTOS 7 WITH GUI 3

2.2 INSTALLING AND CONFIGURING KVM 5
2.2.1 INSTALLING KVM FOR UBUNTU 5
2.2.2 INSTALLING KVM FOR CENTOS 7 5

2.3 INSTALLING AND TESTING DIRECTCORE SOFTWARE 6

2.4 INSTALLING FFMPEG 6
2.4.1 REMOVING PREVIOUS VERSIONS OF YASM, FFMPEG, AND LIBX264 6
2.4.2 DOWNLOAD AND INSTALL THE LATEST VERSIONS OF THE SOFTWARE 7
2.4.3 EDITING CONFIGURATION FILE 9
2.4.4 RUNNING FFMPEG 9

2.5 INSTALLING VLC PLAYER 10
2.5.1 INSTALLING VLC ON UBUNTU 10
2.5.2 INSTALLING VLC ON CENTOS 7 10

2.6 INSTALLING VIRTUAL MACHINES 11
2.6.1 INSTALLING UBUNTU VIRTUAL MACHINE 11
2.6.2 INSTALLING WINDOWS VIRTUAL MACHINE 11
2.6.3 INTERNET ACCESS IN VIRTUAL MACHINES 11

2.7 INSTALLING ROSE AND BOOST 11
2.7.1 INSTALLING ROSE AND BOOST ON CENTOS 7 SYSTEM 11

3 REMOTE DEVICES SOFTWARE INSTALLATION 13

3.1 REMOTE DEVICES OVERVIEW 13

3.2 INSTALLING VIDEO PLAYER 13

4 RUNNING THE VDI DEMO 13

4.1 STREAMTEST DEMO STEPS SEQUENCE 13

# 1 VDI Installation Overview

Signalogic high-performance virtual desktop infrastructure (VDI) is a unique combination of virtual machines (VMs) and multicore high-performance compute (HPC) technology. Traditional approaches to VM deployment present a number of challenges. High hardware costs per machine — along with large size and high power consumption — negate many of the advantages of VM deployment. What's more, the difficulty of programming traditional VM solutions places constraints on engineering resources. High network I/O latency issues often negate gains in computing performance.

Signalogic's High Performance Virtual Machines (HPVMs), embodied in this document as VDI, provide unprecedented economies of scale and efficiency. By utilizing hundreds of compute-intensive multicores in a single 1U server (or thousands in a standard rack), users can realize a dramatic increase in consolidation ratio (the number of virtual machines per server), machine compute power and performance-per-watt ratio. In this way, HPVMs allow infrastructure planners and performance engineers to overcome the traditional high costs and performance issues associated with VMs.

The sections below detail a procedure for the installation of the Signalogic's VDI software package and other software needed for the live streaming of virtual machine desktops to remote Android devices.

# 2 Server Software Installations

## 2.1 Installing Host Linux

Install Ubuntu 14.04 from .iso image available at http://releases.ubuntu.com/14.04/

### 2.1.1 Installing Lightweight GUI for Ubuntu 14.04

Install Lightweight GUI like xfce4 as Ubuntu 14.04 uses Unity 3D, which is based on Compiz and takes a lot of resources.

```
sudo apt-get update
sudo apt-get install xfce4
```

GPL and a graphics card may help with GUI performance, but is not included in the current Signalogic demo setup.

#### *2.1.1.1 9p File System Support*

Ubuntu 14.04 has 9p file system support by default. To verify, execute the command:

```
cat /boot/config-$(uname -r) | grep 9P
```

The above command will print the kernel configuration parameters for the 9p file system:

```
CONFIG_NET_9P=m
CONFIG_NET_9P_VIRTIO=m
CONFIG_NET_9P_RDMA=m
# CONFIG_NET_9P_DEBUG is not set
CONFIG_9P_FS=m
# CONFIG_9P_FSCACHE is not set
CONFIG_9P_FS_POSIX_ACL=y
CONFIG_9P_FS_SECURITY=y
```

## 2.1.2 Installing CentOS 7 with GUI

### *2.1.2.1 Custom Kernel and QEMU for 9p File System Support*

Install all necessary development packages:

```
yum install gcc ncurses ncurses-devel yum-utils rpm-build redhat-rpm-config rpmdevtools
```

Create the Downloads directory, if it does not already exist:

```
cd ~/Downloads
```

Download kernel source from kernel.org with command:

```
wget -c https://www.kernel.org/pub/linux/kernel/v3.x/linux-3.15.6.tar.xz
```

Set environment as the user home directory:

```
cd ~
```

Make sure you are in the home directory at this point. Then, enter the command:

```
rpmdev-setuptree
```

Navigate to the Downloads directory and untar the kernel file:

```
cd ~/Downloads
tar -xJvf linux-3.15.6.tar.xz
cd ~/Downloads/linux-3.15.6/
make menuconfig
```

Turn on modules related to 9p file system:

1. General Setup → Local version
    a. Type a string to identify your kernel version. For example, "***_9fps***".
2. Networking Support → Plan9 Resource Sharing Support (9P2000)
    a. Turn on all modules inside the Plan9 Resource Sharing Support section
3. File systems → Network File Systems → Turn on the following:
    a. Plan9 Resource Sharing Support (9P2000)
    b. 9P POSIX Access Control Lists
    c. 9P Security Labels

Save the configuration and exit.

Build the rpm with command:

```
make rpm
```

Navigate to your ***rpmbuild*** folder with command:

```
cd ~/ rpmbuild/RPMS/x86_64
```

Install your kernel and kernel headers with command:

```
yum localinstall kernel-3.15.6_9pfs-1.x86_64.rpm kernel-headers-
3.15.6_9pfs-1.x86_64.rpm
```

*Note: Names of rpm files may vary.*

The newly installed kernel will have 9p Support.

## 2.2 Installing and Configuring KVM

### 2.2.1 Installing KVM for Ubuntu

Make sure your CPU supports hardware virtualization. To verify whether your CPU supports Intel VT-x or AMD-V, run command:

```
egrep -c '(vmx|svm)'/proc/cpuinfo
```

*If the result is 0, it means that your CPU doesn't support hardware virtualization. If 1 or more, your system does support hardware virtualization. Also, ensure that virtualization is enabled in the BIOS.*

Install the below packages with command:

```
sudo apt-get install qemu-kvm libvirt-bin ubuntu-vm-builder bridge-
utils virt-viewer
```

You can verify if your install has been successful with command:

```
virsh -c qemu:///system list
```

If your install was successful, you will get list of installed virtual machines. If no VMs have been installed, an empty list is returned as shown below:

```
$ virsh -c qemu:///system list
 Id Name                 State
----------------------------------


$
```

Install GUI tool ***virt-manager*** to manage virtual machines:

```
sudo apt-get install virt-manager
```

### 2.2.2 Installing KVM for CentOS 7

If KVM is not already installed, perform the following command:

```
yum -y install qemu-kvm libvirt virt-install bridge-utils
```

## 2.3 Installing and Testing DirectCore Software

Please see the Signalogic DirectCore Reference Guide:

ftp://ftp.signalogic.com/documentation/DirectCore/DirectCore_Reference_Guide_v3_1_0.pdf

## 2.4 Installing FFMPEG

This section provides detailed instructions to remove previous versions, and install the latest versions, of FFMPEG, libx264, and Yasm.

### 2.4.1 Removing Previous Versions of YASM, FFMPEG, and libx264

To remove FFMPEG use commands:

```
apt-get remove libav-tools
apt-get --purge remove libav-tools
apt-get remove libav-ffmpeg
apt-get --purge remove ffmpeg
apt-get autoremove
```

Try to run the FFMPEG command.

It should return an error if it is completely removed from your file system. If FFMPEG command works, that means that FFMPEG is not completely removed from your system.

If you installed FFMPEG using ***make*** commands, in order to uninstall FFMPEG you will have to use one of the following commands to search for FFMPEG files and directories and delete them manually:

```
find / -name 'ffmpeg*'
```

or

```
locate ffmpeg
```

*NOTE: Follow the same procedure to remove Yasm and libx264.*

## 2.4.2 Download and Install the latest Versions of the Software

### *2.4.2.1 Yasm Installation*

To install latest versions of FFMPEG, the current version of Yasm needs to be greater than 1.2.0. To install Yasm, use commands:

*NOTE: Below, "[SPACE]" is not part of the command, rather it emphasizes the need for a single space in its place.*

```
cd ~/Downloads

wget http://www.tortall.net/projects/yasm/releases/yasm-1.3.0.tar.gz

tar xzvf yasm-1.3.0.tar.gz

cd yasm-1.3.0

./configure[SPACE]--prefix="$HOME/ffmpeg_build"[SPACE]--
bindir="$HOME/bin"

make

make install

make distclean
```

### *2.4.2.2 Libx264 Installation*

To install libx264, use commands:

*NOTE: Below, "[SPACE]" is not part of the command, rather it emphasizes the need for a single space in its place.*

```
cd ~/Downloads

wget http://download.videolan.org/pub/x264/snapshots/last_x264.tar.bz2

tar xjvf last_x264.tar.bz2

cd x264-snapshot*

PATH="$PATH:$HOME/bin"[SPACE]./configure[SPACE]--
prefix="$HOME/ffmpeg_build"[SPACE]--bindir="$HOME/bin"[SPACE]--enable-
shared[SPACE]--disable-opencl

PATH="$PATH:$HOME/bin" make

make install

make distclean
```

*Here, libx264 needs to be configured with "--enable-shared" if you are configuring FFMPEG with "--enable-shared" setting.*

#### *2.4.2.3 FFMPEG Installation*

To install FFMPEG, use commands:

*NOTE: Below, "[SPACE]" is not part of the command, rather it emphasizes the need for a single space in its place.*

```
cd ~/Downloads

wget http://ffmpeg.org/releases/ffmpeg-snapshot.tar.bz2

tar xjvf ffmpeg-snapshot.tar.bz2

cd ffmpeg

PATH="$PATH:$HOME/bin"

PKG_CONFIG_PATH="$HOME/ffmpeg_build/lib/pkgconfig"[SPACE]./configure
[SPACE]--prefix="$HOME/ffmpeg_build"[SPACE]--extra-cflags="-
I$HOME/ffmpeg_build/include"[SPACE]--extra-ldflags="-
L$HOME/ffmpeg_build/lib"[SPACE]--bindir="$HOME/bin"[SPACE]--enable-
gpl[SPACE]--enable-shared --enable-libx264[SPACE]--enable-
pthreads[SPACE]--enable-x11grab

PATH="$PATH:$HOME/bin"[SPACE]make

make install

make distclean
```

While trying to ***make*** FFMPEG with '*--enable-x11grab*' configured, if it gives error "*ERROR: Xext not found*" and "*ERROR: Xfixes not found,*"then the ***Xext*** and ***Xfixes*** packages need to be installed.

***Xext*** and ***Xfixes*** packages can be installed for Ubuntu using commands:

```
apt-get install libXext-dev

apt-get install libXfixes-dev
```

***Xext*** and ***Xfixes*** packages can be installed for CentOS using commands:

```
yum install libXext-dev

yum install libXfixes-dev
```

Enter the command below and repeat the above steps again to install FFMPEG.

```
make distclean
```

Installation is necessary to enable the FFMPEG codec library to be used. However, in our case libx264 encoder is used. Therefore, libx264 is separately installed and enabled while configuring FFMPEG. ***PATH*** is provided to store the executable in '*/root/bin*' folder.

### 2.4.3 Editing Configuration File

After installing FFMPEG successfully without any errors, if you try to run FFMPEG from /root/bin folder, an error will be returned:

*"ffmpeg: error while loading shared libraries: libavdevice.so.56: cannot open shared object file."*

The "*loading shared libraries*" error message is received when FFMPEG is not able to locate the file ***libavdevice.so.56***. This happens when either the file is missing from the server, or FFMPEG is not looking at the path the file is located.

So to remove that error, follow the steps:

Open ***ld.so.conf*** file available in **/etc** and add lines:

```
include /etc/ld.so.conf.d/*.conf
/root/ffmpeg_build/lib/
```

After editing the configuration file, run the command:

```
#ldconfig
```

### 2.4.4 Running FFMPEG

Now that the installation is complete and the configuration file has been edited, ensure that you are running FFMPEG as root.

You can run FFMPEG from **/root/bin** folder as follows:

```
./ffmpeg -h
```

This will show all the codecs and settings that are enabled during FFMPEG configuration.

## 2.5 Installing VLC Player

### 2.5.1 Installing VLC on Ubuntu

VLC player can be installed using the command below, or from Ubuntu Software Centre available on Ubuntu GUI.

```
apt-get update
apt-get install vlc
```

### 2.5.2 Installing VLC on CentOS 7

To install EPEL and LinuxRO repository, use commands:

```
#yum install epel-release
#yum localinstall –nogpgcheck
http://li.nux.ro/download/nux/dextop/el7/x86_64/nux-dextop-release-0-1.el7.nux.noarch.rpm
```

Enable EPEL and LinuxRO repository using commands:

```
#yum clean all
#yum update
```

To install VLC Media Player, use commands:

```
#yum install vlc
```

To verify Installation of VLC Media Player, use commands:

```
#exit
$vlc
```

*Note: you are now ready to use VLC on your system.*

## 2.6 Installing Virtual Machines

### 2.6.1 Installing Ubuntu Virtual Machine

For Ubuntu, Install an Ubuntu virtual machine, add a network bridge, and install FFMPEG.

### 2.6.2 Installing Windows Virtual Machine

For Windows, Install a Windows virtual machine, setup remote desktop sharing, install FFMPEG, and install Windows Sync Utility.

### 2.6.3 Internet Access in Virtual Machines

Ensure that your virtual machine has internet access.

## 2.7 Installing ROSE and Boost

The following section describes installation of Boost 1.45 version and rose 0.9.5a version.

### 2.7.1 Installing ROSE and Boost on CentOS 7 System

Install gcc44 and g++44. Create softlinks gcc and g++ to gcc44 and g++44. Run yum install glibc-static, if it is not already installed.

Add the following to the file, /root/boost_1_45_0/tools/build/v2/user-config.jam, which forces boost compilation to use g++44 compiler

```
using gcc : 44 : /usr/bin/g++44 ;
```

#### *2.7.1.1 Boost Installation:*

Download “boost_1_45_0.tar.bz2” from:

http://sourceforge.net/projects/boost/files/boost/1.45.0/

Untar the file:

```
tar -xjvf boost_1_45_0.tar.bz2
```

Apply patch lines to before enum xtime_clock_types line in

```
/root/boost_1_45_0/boost/thread/xtime.hpp file
```

(Assuming untar boost_1_45_0 is in /root folder)

```
#ifdef TIME_UTC
      #undef TIME_UTC
```

```
#endif
```

```
./bootstrap.sh --prefix=/root/boostInstall
```

```
./bjam install  --prefix=/root/boostInstall
```

Navigate to BOOST_INSTALL directory and verify libboost_wave.a library is built.

#### *2.7.1.2 ROSE Installation*

Untar the file:

```
tar -xjvf rose-0.9.5a-without-EDG-19322.tar.gz
```

Create rose install folder:

```
mkdir /root/roseInstall
```

Navigate to ***roseInstall***:

```
cd /root/roseInstall
```

Enter command:

```
/root/rose-0.9.5a-19322/configure --prefix=/root/roseInstall --with-
boost=/root/boostInstall/ --with-boost-libdir=/root/boostInstall/lib -
-without-java --with-boost-wave
```

Navigate to:

```
/root/roseInstall/projects/simulator/Makefile
```

Within ***Makefile***, change LDFLAGS to:

```
LDFLAGS = -L/usr/lib64/ -lrt -lpthread
```

Enter command:

```
make
make install
```

# 3 Remote Devices Software Installation

## 3.1 Remote Devices Overview

The Signalogic VDI software package is compatible with Windows and Android.

## 3.2 Installing Video Player

Install MPlayer on your Windows Surface or Android tablet.

# 4 Running the VDI Demo

## 4.1 streamTest demo Steps Sequence

When server first boots there is a start.sh script under path /root using command:

```
source start.sh
```

The script will load the driver, create shared folder, and assign permissions to it. The user will also be taken to the streamTest folder.

Start both VMs. On HOST, use the following command:

```
virt-manager
```

A graphical view of installed VMs will appear. Log into VMs by entering credentials.

On Both VMs, run the script start.sh located under path: /root using command:

```
source start.sh
```

The script will mount the 9P fileystem, create a shared-file folder on VM RAM disk, and takes user in /root/bin folder to run ffmpeg command.

On HOST, run the streamTest using command:

```
./streamTest -m0x303 -f1600 -estream.out -cSIGC66XX-64 -i -
/mnt/ramdisk/share/screen4_1.yuv -x1280 -y720 -s2 -r30 -
D192.168.1.64:45056:60-af-6d-75-75-f1 -B1500000 -i -
/mnt/ramdisk/share/screen5_1.yuv -x1280 -y720 -s2 -r30 -
D192.168.1.62:45056:b4-ce-f6-9d-3f-36 -s2 -B1500000
```

Here,

- m0x303 = It shows how many CPUs and how many cores are being used. In this case 2 CPUs and 2 cores on each CPUs are being used.

- f1600 = clock rate of 1.6 GHz
- cSIGC66XX-64 = Octal card is being used.
- i -/mnt/ramdisk/share/screen4_1.yuv = input yuv file
- x1280, -y720 = Resolution parameters
- s2 = Continuous mode
- r = Frame rate
- D = IP Address:Port:MAC Address
- B = Bit rate

Once streamTest is started, it will wait on the yuv file that has been provided as input.

Enter FFMPEG command on both of VMs simultaneously that will capture the VM screen and will write that yuv file into shared memory region between host and VMs.

Example command:

```
./ffmpeg -s 1280x720 -framerate 30 -f x11grab -i :0.0+0,0 -pix_fmt
yuv420p -threads 2 -framerate 30 -f rawvideo -t 30 -vsync 0
/tmp/host_files/screen4_1.yuv
```

Once the FFMPEG starts capturing the VM screen, streamTest will find the .yuv file in shared folder and will start encoding it in H.264 and will stream it over the network.

Start MPlayer on tablet, then enter URL of sdp file location that will be used to receive RTP stream by the MPlayer.

URL: http://signalogic.com/sdp/br60.sdp

After streamTest stops, delete the generated screen1.yuv and screen2.yuv files in shared folder on both VMs before restarting the stream.